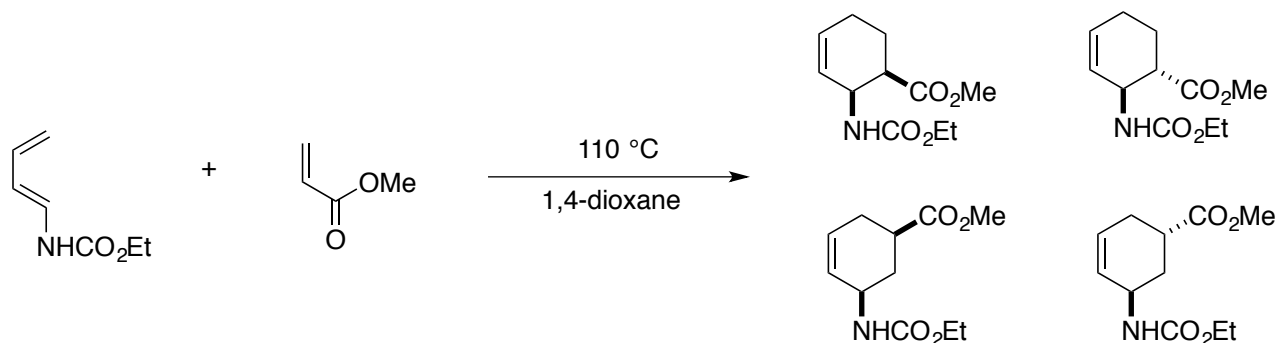


Chem 106: Computational Handout 4 Selectivity in a Diels–Alder Reaction

Objectives: practice with transition states, composite single points, evaluating selectivity

In This Exercise: we will consider this Diels–Alder reaction (Overman *JACS* **1978** *100* 3182)



62% total yield
>100:1 *ortho:meta*
81:19 *endo:exo*

This reaction was studied computationally by Houk at the time, but with limited tools he could only correlate ionization potentials to selectivity. Here we apply a modern DFT method and ask whether it can reproduce the observed selectivities directly.

Now that you are more familiar with calculations, these instructions will be less verbose.

Setup

As usual, update your `chem106_calculations` repository both locally and on Odyssey using `git pull`. The repository contains my files for this exercise, but you should only look at them if you get stuck.

Draw Truncated Structures, Consider Rotamers

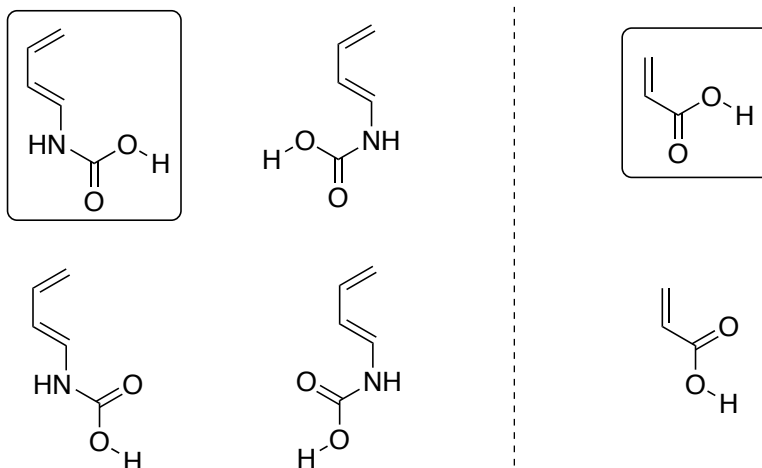
To start, we will consider the following truncated system:



Because DFT calculations square as N^4 , where N is the number of basis functions, removing a small number of atoms can speed up the initial calculations dramatically. The carbamic acid might not be a realistic laboratory analog, but will behave almost identically *in silico*. (We will consider the full structure later in this exercise.)

We have to think about which rotamers to use. The easiest one is the E/Z rotamers about the carboxylic acids. The Z conformation drawn above is decisively lower in energy, so there is no need to consider the E conformation. Similarly, the diene can only react in an s-cis conformation, so that is the only one that needs to be considered.

The tricky decisions are about the following rotamers:



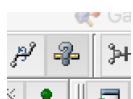
For simplicity, we will only consider the indicated rotamers. (A more in-depth analysis would be required for the project or a publication.) We will start by drawing four structures in which the diene and dienophile are frozen in close proximity, as a prelude to finding the four transition states. Since we only need to determine the regioselectivity and endo/exo selectivity, there is no need to locate isolated structures of the starting materials or products.

1. Draw the structures corresponding to the transition states discussed above. You should have four structures, corresponding to all possible combinations of ortho/meta and endo/exo.

Use these filenames:

```
da_meta_endo-b31yp_d3bj-midix-freeze.gjf
da_meta_exo-b31yp_d3bj-midix-freeze.gjf
da_ortho_endo-b31yp_d3bj-midix-freeze.gjf
da_ortho_exo-b31yp_d3bj-midix-freeze.gjf
```

When drawing the structures, place the diene and dienophile about 2.1–2.3 Å apart, as measured by the forming C–C bond distances. Do not let any of the atoms get too close together. You can tell how close two atoms are by using the “question mark” tool:



If this isn't feasible, place them farther apart, perform the optimization, and then move them closer together manually.

Alert! In general, you should try to preserve the ordering of atoms in sets of related files. This makes subsequent analysis easier. Draw one structure in which the diene and dienophile are together, and then use bond rotations and successive save commands to generate the other files.

Alert! The filename system here is arbitrary, but I suggest you follow something very systematic. It is easy to get lost with thousands of files. Do not use notebook numbers. This ends up being extremely cryptic. (Imagine a library where all books are only labeled by the date they were published.)

2. Setup the input files to do a B3LYP-D3(BJ)/MIDI!/PCM frozen optimization:

```
%nprocshared=4
%mem=3GB
#p b3lyp empiricaldispersion=gd3bj midix scrf=(pcm,solvent=1,4-dioxane)
opt=modredundant freq=noraman
```

B3LYP is a DFT method that is well-established to be good for pericyclic reactions. We also know we need dispersion corrections. MIDI! is a minimal basis set (that we will upgrade shortly). PCM is a standard solvent model.

Don't forget to add the modredundant coordinate at the end of the file, separated from the geometry by a blank line. Add blank lines at the end of the file as well. It should look like this:

```
O          2.08297646   -1.10198483    1.26394306
H          -4.53710300   -1.21631900   -0.40086600
H          2.89937146   -1.36053483    1.77352006
```

```
B 8 17 F
```

Freeze the bond that is likely to be shorter (use your chemical intuition for an asynchronous reaction) and let the other forming bond distance be optimized.

3. Run the optimizations. Send to `serial_requeue` or `general` and give it a couple of hours. Be sure to give the job an appropriate amount of memory.

4. Make sure each one converges. If you see an imaginary frequency corresponding to the Diels–Alder that is greater in magnitude than 100 cm^{-1} , then proceed to the next section. Don't worry if you see some other small imaginary frequencies. These will be optimized out.

If you don't see the imaginary, bring the structures closer to 2.1 \AA . You could scan this distance, but these transition states can be found much faster by following the procedure here.

(It is often helpful to know some typical ground and transition state bond lengths. A C–C bond is about 1.55 \AA in length, but stretches to over 2.0 \AA in a transition state.)

Locate Transition State

1. Save the last geometry from each frozen optimization as the input files:

```
da_meta_endo-b3lyp_d3bj-midix-ts.gjf
da_meta_exo-b3lyp_d3bj-midix-ts.gjf
da_ortho_endo-b3lyp_d3bj-midix-ts.gjf
da_ortho_exo-b3lyp_d3bj-midix-ts.gjf
```

2. In the input file, release the constraints and call for a transition state optimization:

```
%nprocshared=4
%mem=3GB
#p b3lyp empiricaldispersion=gd3bj midix scrf=(pcm,solvent=1,4-dioxane)
opt=(ts,calcfc,maxstep=5,noeigentest) freq=noraman
```

3. Download the structures. If they converged to a transition state, congratulations! If they didn't, check the optimization graph, and find the point with the highest energy and the lowest gradient and re-submit it until it does converge.

Increase Basis Set Size

Increase the basis set size to 6-31G* and find the new transition states. Call the files:

```
da_meta_endo-b3lyp_d3bj-631gd-ts.gjf
da_meta_exo-b3lyp_d3bj-631gd-ts.gjf
da_ortho_endo-b3lyp_d3bj-631gd-ts.gjf
da_ortho_exo-b3lyp_d3bj-631gd-ts.gjf
```

You will see that the transition state barely changes in geometry. There is no need to freeze/scan.

Find Full Transition States

1. Add the ethyl group on the carbamate and the methyl group on the acrylate. To preserve the atom ordering, add the groups with the exact same sequence of operations on each file.

2. Locate the B3LYP-D3(BJ)/6-31G*/PCM structures for each of these new files:

```
da_meta_endo_full-b3lyp_d3bj-631gd-ts.gjf
da_meta_exo_full-b3lyp_d3bj-631gd-ts.gjf
da_ortho_endo_full-b3lyp_d3bj-631gd-ts.gjf
da_ortho_exo_full-b3lyp_d3bj-631gd-ts.gjf
```

3. Increase the basis set to cc-pVDZ and locate the new transition states:

```
da_meta_endo_full-b3lyp_d3bj-dz-ts.gjf
da_meta_exo_full-b3lyp_d3bj-dz-ts.gjf
da_ortho_endo_full-b3lyp_d3bj-dz-ts.gjf
da_ortho_exo_full-b3lyp_d3bj-dz-ts.gjf
```

Specify the reaction temperature:

```
%nprocshared=4
%mem=3GB
#p b3lyp empiricaldispersion=gd3bj cc-pVDZ scrf=(pcm,solvent=1,4-dioxane)
opt=(ts,calcfc,maxstep=5,noeigentest) freq=noraman temperature=383
```

We will need the Gibbs free energy later at this temperature.

Perform Single Points

At cc-pVDZ, the geometries will be nearly converged (for our purposes) to their complete basis set (CBS) limit. However, their *energies* will suffer from significant BSIE and BSSE.

1. Compute *single point energies* using the cc-pVDZ geometries at cc-pVTZ:

```
%nprocshared=4
%mem=3GB
#p b3lyp empiricaldispersion=gd3bj cc-pVTZ scrf=(pcm,solvent=1,4-dioxane)
```

A triple-zeta basis set increases the number of basis functions considerably (by how much?) so these calculations could take several hours (just for one point!). This route card is a single point calculation because it does not contain an `opt` statement.

Use these filenames:

```
da_meta_endo_full-b3lyp_d3bj-dz-ts-tz_sp.gjf
da_meta_exo_full-b3lyp_d3bj-dz-ts-tz_sp.gjf
da_ortho_endo_full-b3lyp_d3bj-dz-ts-tz_sp.gjf
da_ortho_exo_full-b3lyp_d3bj-dz-ts-tz_sp.gjf
```

Alert! Frequency analysis is only valid at a true stationary point (i.e., where the gradient is numerically zero). It would not be appropriate to try and calculate the frequencies of these structures because they are stationary points at cc-pVDZ, but not cc-pVTZ.

2. Repeat this procedure at cc-pVQZ. These jobs will take at least overnight. (If you don't get to this stage, use the files I have provided.)

```
da_meta_endo_full-b3lyp_d3bj-dz-ts-qz_sp.gjf
da_meta_exo_full-b3lyp_d3bj-dz-ts-qz_sp.gjf
da_ortho_endo_full-b3lyp_d3bj-dz-ts-qz_sp.gjf
da_ortho_exo_full-b3lyp_d3bj-dz-ts-qz_sp.gjf
```

Analysis

I will show you how to use Linux to extract the energies quickly here. If you feel that it is too complicated to do things this way, feel free to look at the files manually.

1. Extract the cc-pVDZ electronic energies .

First, list the relevant files with a loop:

```
~/chem106_calculations/diels_alder/out $ for i in *full-b3lyp_d3bj-dz-ts.out; do echo $i; done
da_meta_endo_full-b3lyp_d3bj-dz-ts.out
da_meta_exo_full-b3lyp_d3bj-dz-ts.out
da_ortho_endo_full-b3lyp_d3bj-dz-ts.out
da_ortho_exo_full-b3lyp_d3bj-dz-ts.out
```

Semicolons (;) allow multiple commands to be placed on one line. \$i represents the name of the file each time we go through the loop.

Then, use `grep` and `tail` to find the last electronic energy:

```
~/chem106_calculations/diels_alder/out $
for i in *full-b3lyp_d3bj-dz-ts.out; do echo $i; grep "SCF Done" $i | tail -n1 ; done

da_meta_endo_full-b3lyp_d3bj-dz-ts.out
  SCF Done:  E(RB3LYP) =  -785.097182318      A.U. after  1 cycles
da_meta_exo_full-b3lyp_d3bj-dz-ts.out
  SCF Done:  E(RB3LYP) =  -785.097175646      A.U. after  1 cycles
da_ortho_endo_full-b3lyp_d3bj-dz-ts.out
  SCF Done:  E(RB3LYP) =  -785.110802524      A.U. after  1 cycles
da_ortho_exo_full-b3lyp_d3bj-dz-ts.out
  SCF Done:  E(RB3LYP) =  -785.107553904      A.U. after  1 cycles
```

For clarity, I put the prompt and command on different lines. You should not press enter before the command. Remember, `grep` searches for a line of text in a given file (\$i) and we use a pipe (|) to send the result to `tail`, which is asked for the last line (-n1).

Now, let's do away with the filenames (this will make it easier to cut and paste into a spreadsheet):

```
~/chem106_calculations/diels_alder/out $
for i in *full-b3lyp_d3bj-dz-ts.out; do grep "SCF Done" $i | tail -n1 ; done
SCF Done:  E(RB3LYP) = -785.097182318      A.U. after 1 cycles
SCF Done:  E(RB3LYP) = -785.097175646      A.U. after 1 cycles
SCF Done:  E(RB3LYP) = -785.110802524      A.U. after 1 cycles
SCF Done:  E(RB3LYP) = -785.107553904      A.U. after 1 cycles
```

Let's extract only the energies (in hartree):

```
~/chem106_calculations/diels_alder/out $
for i in *full-b3lyp_d3bj-dz-ts.out; do grep "SCF Done" $i | tail -n1 ; done | awk
'{print $5}'
-785.097182318
-785.097175646
-785.110802524
-785.107553904
```

This pipes the entire output of the loop to `awk`, which is asked to print the fifth field (`$5`). Finally, calculate the relative energies in kcal/mol:

```
~/chem106_calculations/diels_alder/out $
for i in *full-b3lyp_d3bj-dz-ts.out; do grep "SCF Done" $i | tail -n1 ; done | awk
'{print ($5+785.110802524)*627.509469}'
8.54681
8.55099
0
2.03854
```

Again, the whole command should be on one line. This means to take the energy, add the lowest energy (by copying and pasting from the last result), and apply the hartree-to-kcal conversion factor.

2. Extract the cc-pVDZ free energies and the cc-pVTZ and cc-PVQZ electronic energies. Invent your own one-liners! Feel free to do things in a somewhat different way. There's no one right answer. For example, if you want to calculate the relative energies in a spreadsheet, you can use this more concise one-liner:

```
~/chem106_calculations/diels_alder/out $ awk '/SCF Done/ {print $5}' *tz*.out
-785.351963380
-785.352087518
-785.363240922
-785.360927131
```

This uses `awk` to search for `SCF Done` in each file and print out only the energy. (This method is more concise, but doesn't work for finding the last energy of each file.)

3. What are the selectivity predictions using the DZ, TZ, and QZ electronic energies?

4. What are the selectivity predictions using the DZ, TZ, and QZ electronic energies + the DZ Gibbs correction?

Alert! My answers are on the next page, but don't look until you're done!

Discussion

My selectivity predictions are:

(kcal/mol)	DZ electronic	TZ electronic	QZ electronic	DZ (elec) + DZ (thermal)	TZ (elec) + DZ (thermal)	QZ (elec) + DZ (thermal)
Meta, endo	8.5	7.1	6.6	6.2	5.3	5.1
Meta, exo	8.6	7.0	6.5	5.6	4.6	4.4
Ortho, endo	0.0	0.0	0.0	0.0	0.6	0.8
Ortho, exo	2.0	1.5	1.2	0.0	0.0	0.0

These are “single point” energies that would be reported as something like:

B3LYP-D3(BJ)/cc-pVQZ/PCM // B3LYP-D3(BJ)/cc-pVDZ/PCM

meaning cc-pVQZ single point energies on cc-pVDZ energies.

The regioselectivity (>100:1 ortho) is correctly predicted with all methods. This is also the same result you would predict from resonance structures.

However, the endo/exo selectivity (experimental 4:1) is very poorly predicted. Using electronic energies only, the endo product is predicted. The predicted selectivity decreases with basis set size, presumably because there is more BSSE in the endo transition state.

With the thermal correction, the predictions are completely overturned! The exo product is now predicted to be favored by 0.8 kcal/mol, contrary to experiment. Endo/exo selectivity predictions are known to be problematic. The source of the problem is unknown—it could be electron correlation/dispersion, or it could be subtle solvent effect that is not treated well by PCM. It is also possible that the prediction would be better with a full consideration of all rotamers (but I doubt it).

*Eugene Kwan
March 2017*